```
<xsd:element name="purchaseOrder" type="POType1"/>
<xsd:element name="comment" type="xsd:string"/>

<xsd:complexType name="POType1">
  <xsd:sequence>
    <xsd:element name="shipTo" type="USAddress"/>
    <xsd:element name="billTo" type="USAddress" minOccurs="0"/>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="items"/>
  </xsd:sequence>
</xsd:complexType>
```

(a)

```
<xsd:element name="purchaseOrder" type="POType2"/>
<xsd:element name="comment" type="xsd:string"/>

<xsd:complexType name="POType2">
  <xsd:sequence>
    <xsd:element name="shipTo" type="USAddress"/>
    <xsd:element name="billTo" type="USAddress"/>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="items"/>
  </xsd:sequence>
</xsd:complexType>
```

(b)

Figure 1. Definition of purchaseOrder **element in (a) Source Schema (b) Target Schema. The figure shows a fragment of each schema.**

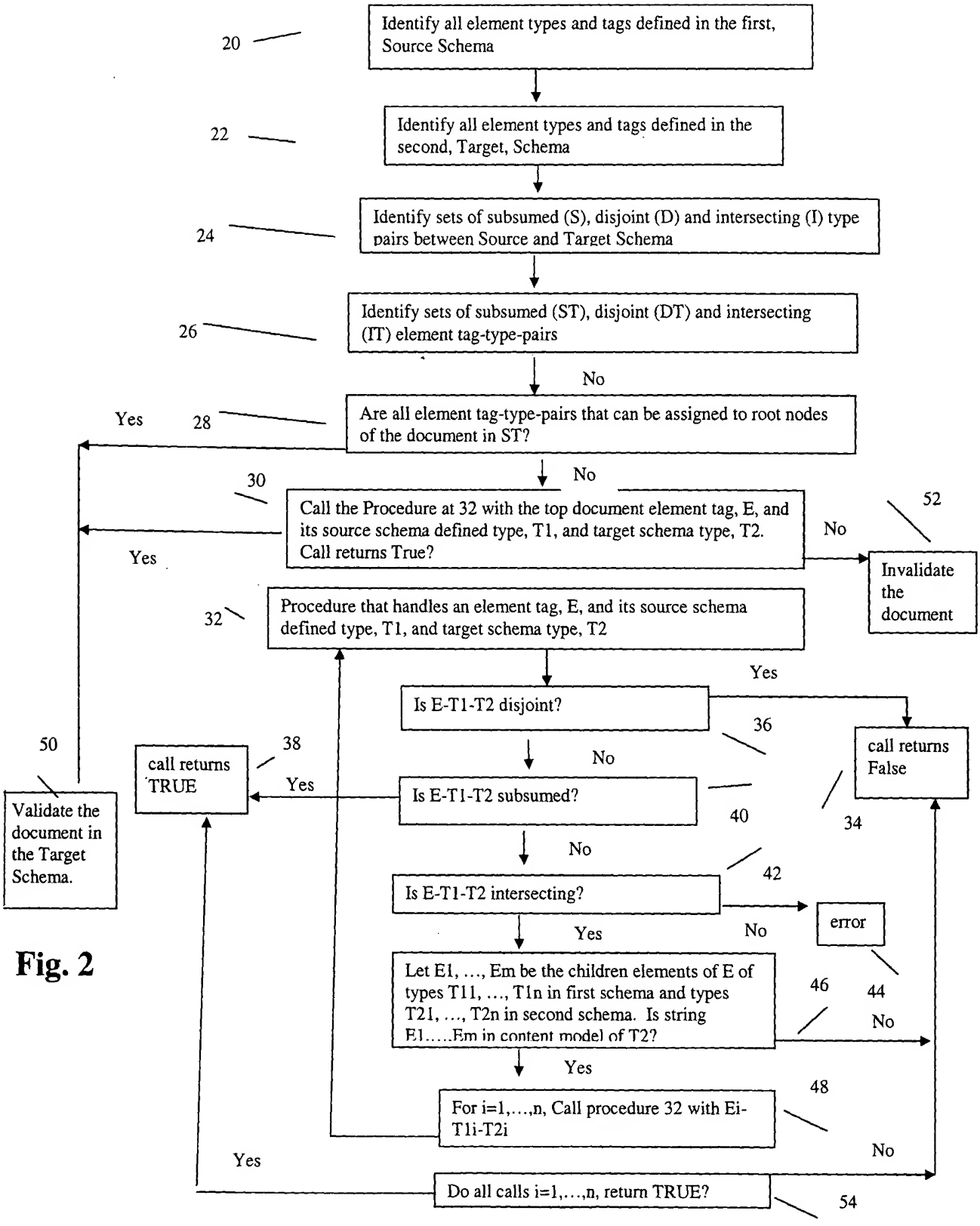
20
Identify all element types and tags defined in the first, Source Schema
22
Identify all element types and tags defined in the second, Target, Schema
24
Identify sets of subsumed (S), disjoint (D) and intersecting (I) type pairs between Source and Target Schema
26
Identify sets of subsumed (ST), disjoint (DT) and intersecting (IT) element tag-type-pairs
No
Yes
28
Are all element tag-type-pairs that can be assigned to root nodes of the document in ST?
No
30
Call the Procedure at 32 with the top document element tag, E, and its source schema defined type, T1, and target schema type, T2. Call returns True?
No
52
Invalidate the document
Yes
32
Procedure that handles an element tag, E, and its source schema defined type, T1, and target schema type, T2
Yes
Is E-T1-T2 disjoint?
36
call returns False
No
50
38
call returns TRUE
Yes
Is E-T1-T2 subsumed?
40
34
Validate the document in the Target Schema.
No
42
Is E-T1-T2 intersecting?
error
Yes
No
Let E1, ..., Em be the children elements of E of types T11, ..., T1n in first schema and types T21, ..., T2n in second schema. Is string E1.....Em in content model of T2?
46
44
No
Yes
48
For i=1,...,n, Call procedure 32 with Ei-T1i-T2i
No
Yes
Do all calls i=1,...,n, return TRUE?
54

Fig. 2

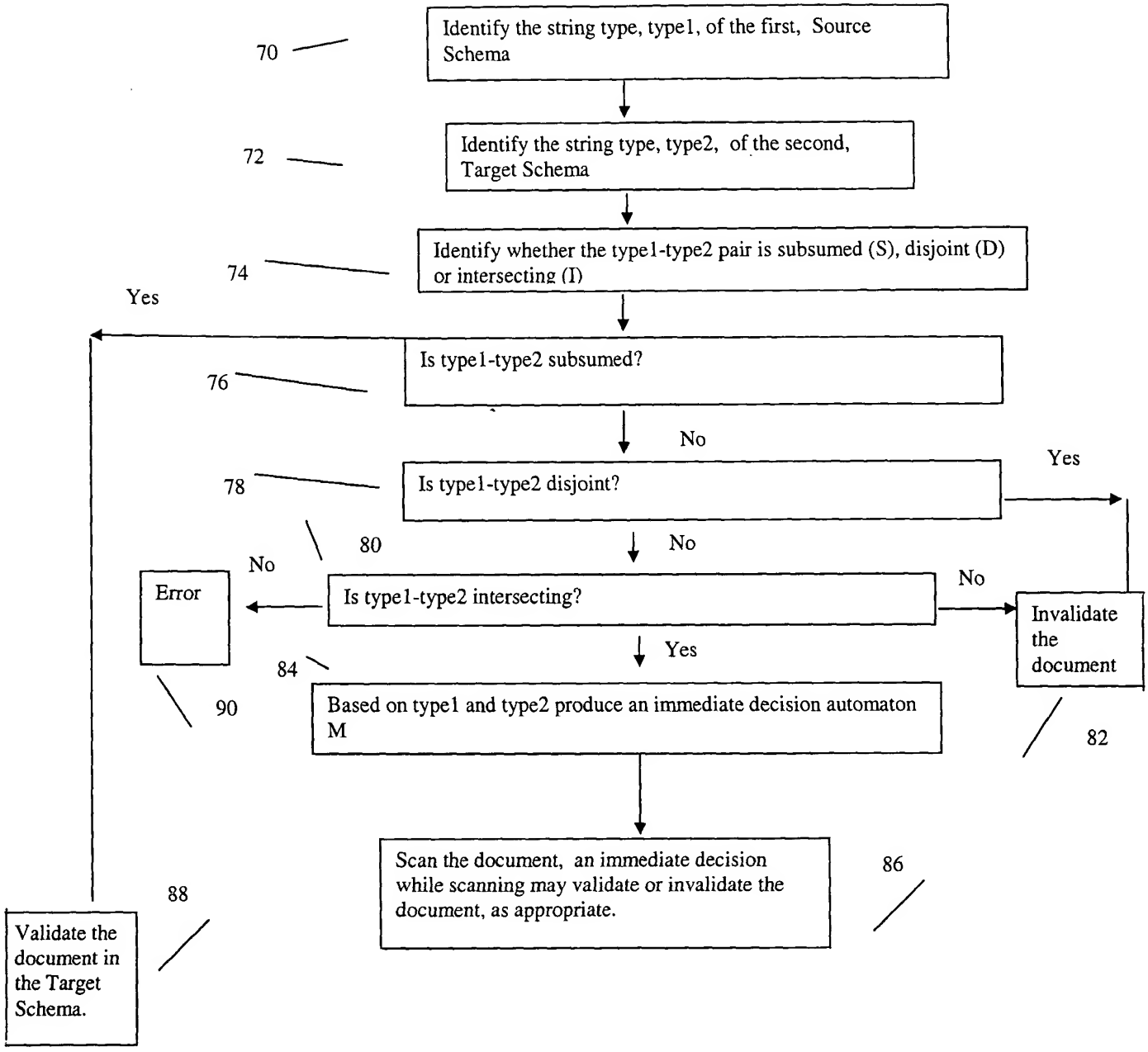
Identify the string type, type1, of the first, Source Schema
70
Identify the string type, type2, of the second, Target Schema
72
Identify whether the type1-type2 pair is subsumed (S), disjoint (D) or intersecting (I)
74
Yes
Is type1-type2 subsumed?
76
No
Is type1-type2 disjoint?
78
Yes
80
No
No
Error
Is type1-type2 intersecting?
No
Invalidate the document
Yes
84
90
Based on type1 and type2 produce an immediate decision automaton M
82
Scan the document, an immediate decision while scanning may validate or invalidate the document, as appropriate.
86
88
Validate the document in the Target Schema.

Fig. 3

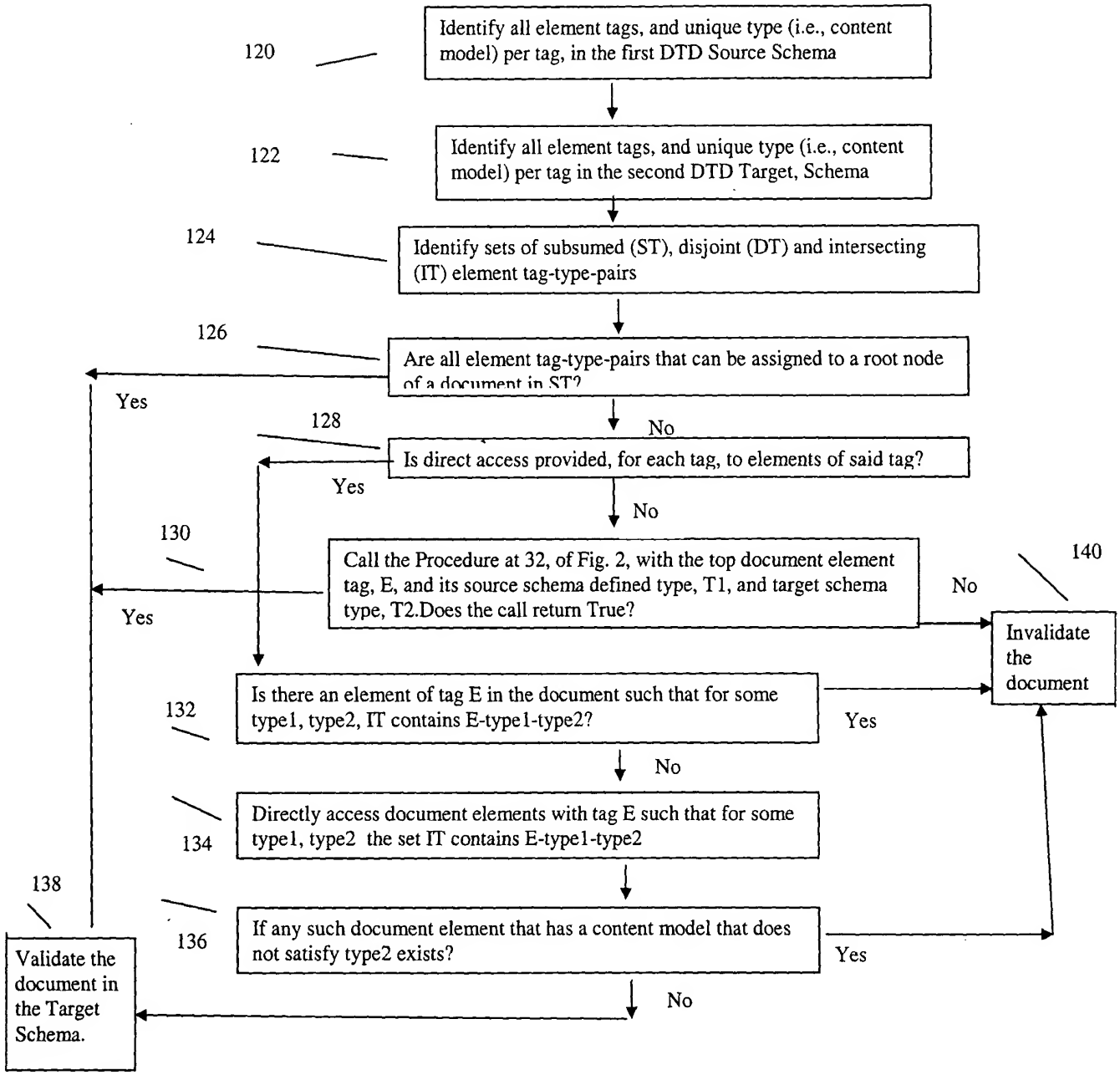
120
Identify all element tags, and unique type (i.e., content model) per tag, in the first DTD Source Schema
122
Identify all element tags, and unique type (i.e., content model) per tag in the second DTD Target, Schema
124
Identify sets of subsumed (ST), disjoint (DT) and intersecting (IT) element tag-type-pairs
126
Are all element tag-type-pairs that can be assigned to a root node of a document in ST?
Yes
No
128
Is direct access provided, for each tag, to elements of said tag?
Yes
No
130
Call the Procedure at 32, of Fig. 2, with the top document element tag, E, and its source schema defined type, T1, and target schema type, T2.Does the call return True?
Yes
No
140
Invalidate the document
132
Is there an element of tag E in the document such that for some type1, type2, IT contains E-type1-type2?
Yes
No
134
Directly access document elements with tag E such that for some type1, type2 the set IT contains E-type1-type2
138
136
If any such document element that has a content model that does not satisfy type2 exists?
Yes
No
Validate the document in the Target Schema.

Fig. 4

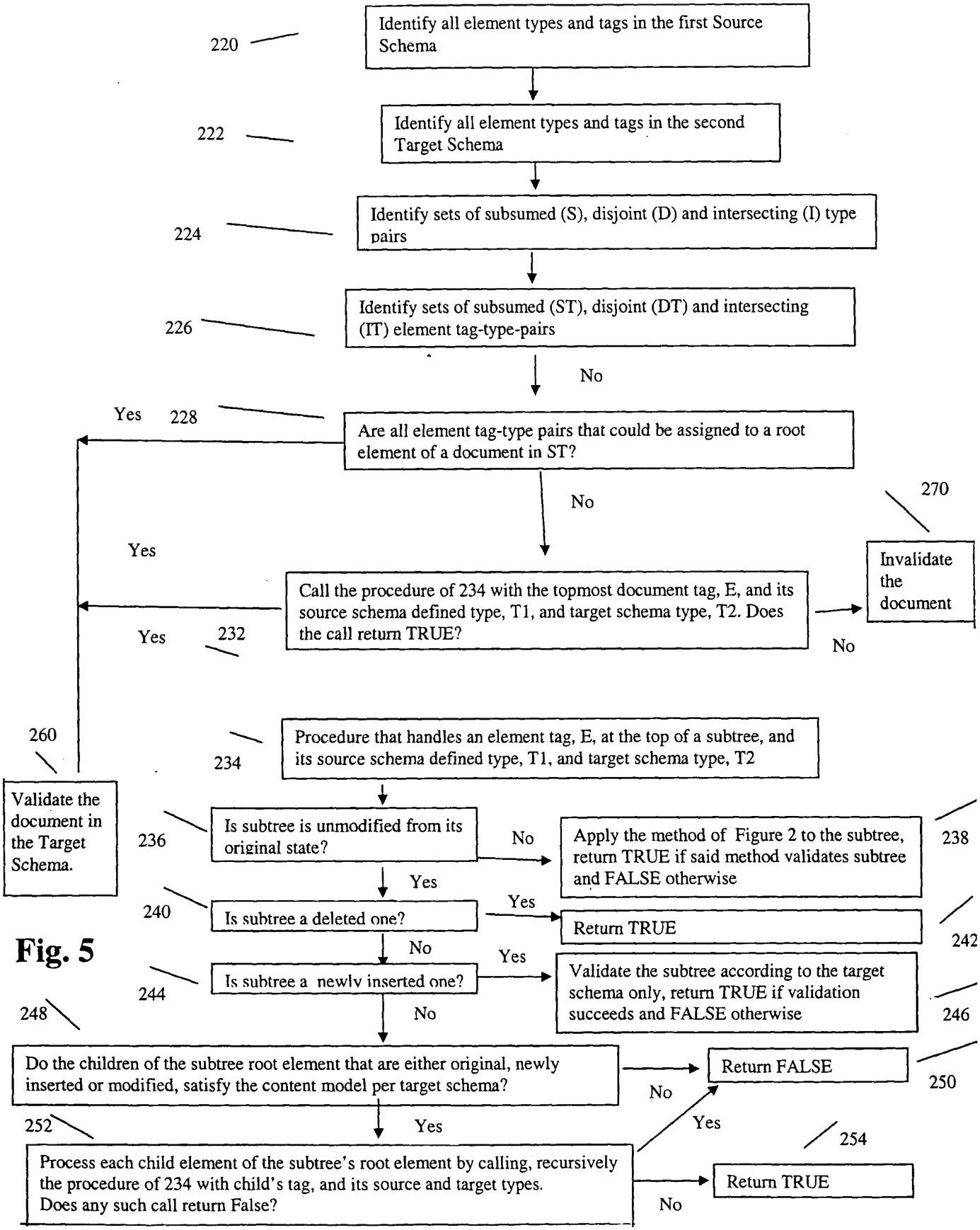
220
Identify all element types and tags in the first Source Schema
222
Identify all element types and tags in the second Target Schema
224
Identify sets of subsumed (S), disjoint (D) and intersecting (I) type pairs
226
Identify sets of subsumed (ST), disjoint (DT) and intersecting (IT) element tag-type-pairs
No
Yes
228
Are all element tag-type pairs that could be assigned to a root element of a document in ST?
No
270
Invalidate the document
Yes
Call the procedure of 234 with the topmost document tag, E, and its source schema defined type, T1, and target schema type, T2. Does the call return TRUE?
No
Yes
232
260
Validate the document in the Target Schema.
234
Procedure that handles an element tag, E, at the top of a subtree, and its source schema defined type, T1, and target schema type, T2
236
Is subtree is unmodified from its original state?
No
Apply the method of Figure 2 to the subtree, return TRUE if said method validates subtree and FALSE otherwise
238
Yes
240
Is subtree a deleted one?
Yes
Return TRUE
242
No
Yes
244
Is subtree a newly inserted one?
Validate the subtree according to the target schema only, return TRUE if validation succeeds and FALSE otherwise
246
248
No
Do the children of the subtree root element that are either original, newly inserted or modified, satisfy the content model per target schema?
Return FALSE
250
No
Yes
252
Yes
254
Process each child element of the subtree's root element by calling, recursively the procedure of 234 with child's tag, and its source and target types. Does any such call return False?
Return TRUE
No

Fig. 5

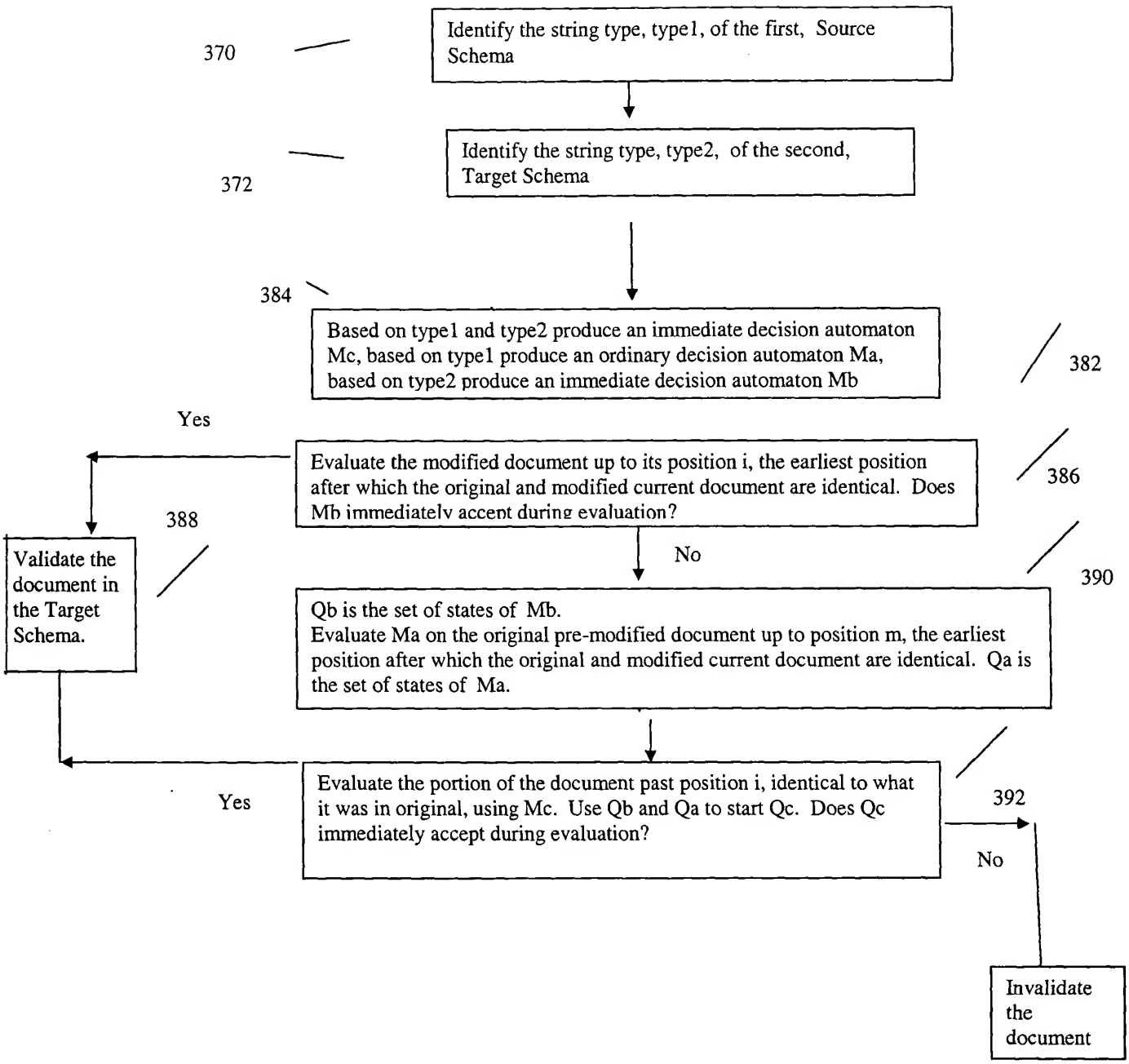
370
Identify the string type, type1, of the first, Source Schema
372
Identify the string type, type2, of the second, Target Schema
384
Based on type1 and type2 produce an immediate decision automaton Mc, based on type1 produce an ordinary decision automaton Ma, based on type2 produce an immediate decision automaton Mb
382
Yes
Evaluate the modified document up to its position i, the earliest position after which the original and modified current document are identical. Does Mb immediately accept during evaluation?
386
388
Validate the document in the Target Schema.
No
390
Qb is the set of states of Mb.
Evaluate Ma on the original pre-modified document up to position m, the earliest position after which the original and modified current document are identical. Qa is the set of states of Ma.
Yes
Evaluate the portion of the document past position i, identical to what it was in original, using Mc. Use Qb and Qa to start Qc. Does Qc immediately accept during evaluation?
392
No
Invalidate the document
No

Fig. 6

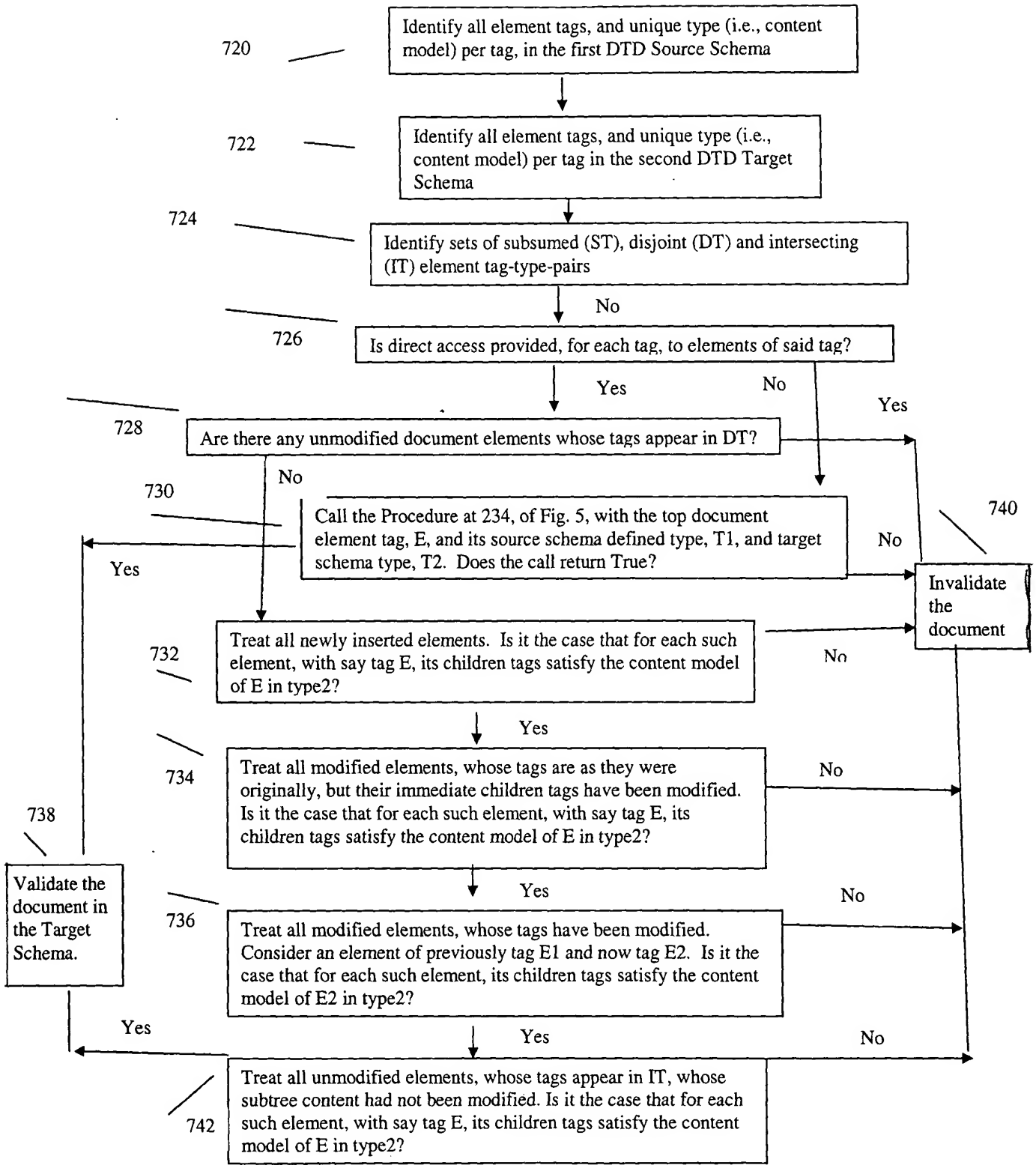
720
Identify all element tags, and unique type (i.e., content model) per tag, in the first DTD Source Schema
722
Identify all element tags, and unique type (i.e., content model) per tag in the second DTD Target Schema
724
Identify sets of subsumed (ST), disjoint (DT) and intersecting (IT) element tag-type-pairs
726
Is direct access provided, for each tag, to elements of said tag?
No
Yes
No
Yes
728
Are there any unmodified document elements whose tags appear in DT?
No
730
Call the Procedure at 234, of Fig. 5, with the top document element tag, E, and its source schema defined type, T1, and target schema type, T2. Does the call return True?
Yes
No
740
Invalidate the document
732
Treat all newly inserted elements. Is it the case that for each such element, with say tag E, its children tags satisfy the content model of E in type2?
No
Yes
734
Treat all modified elements, whose tags are as they were originally, but their immediate children tags have been modified. Is it the case that for each such element, with say tag E, its children tags satisfy the content model of E in type2?
No
738
Validate the document in the Target Schema.
Yes
736
Treat all modified elements, whose tags have been modified. Consider an element of previously tag E1 and now tag E2. Is it the case that for each such element, its children tags satisfy the content model of E2 in type2?
No
Yes
Yes
No
742
Treat all unmodified elements, whose tags appear in IT, whose subtree content had not been modified. Is it the case that for each such element, with say tag E, its children tags satisfy the content model of E in type2?

Fig. 7

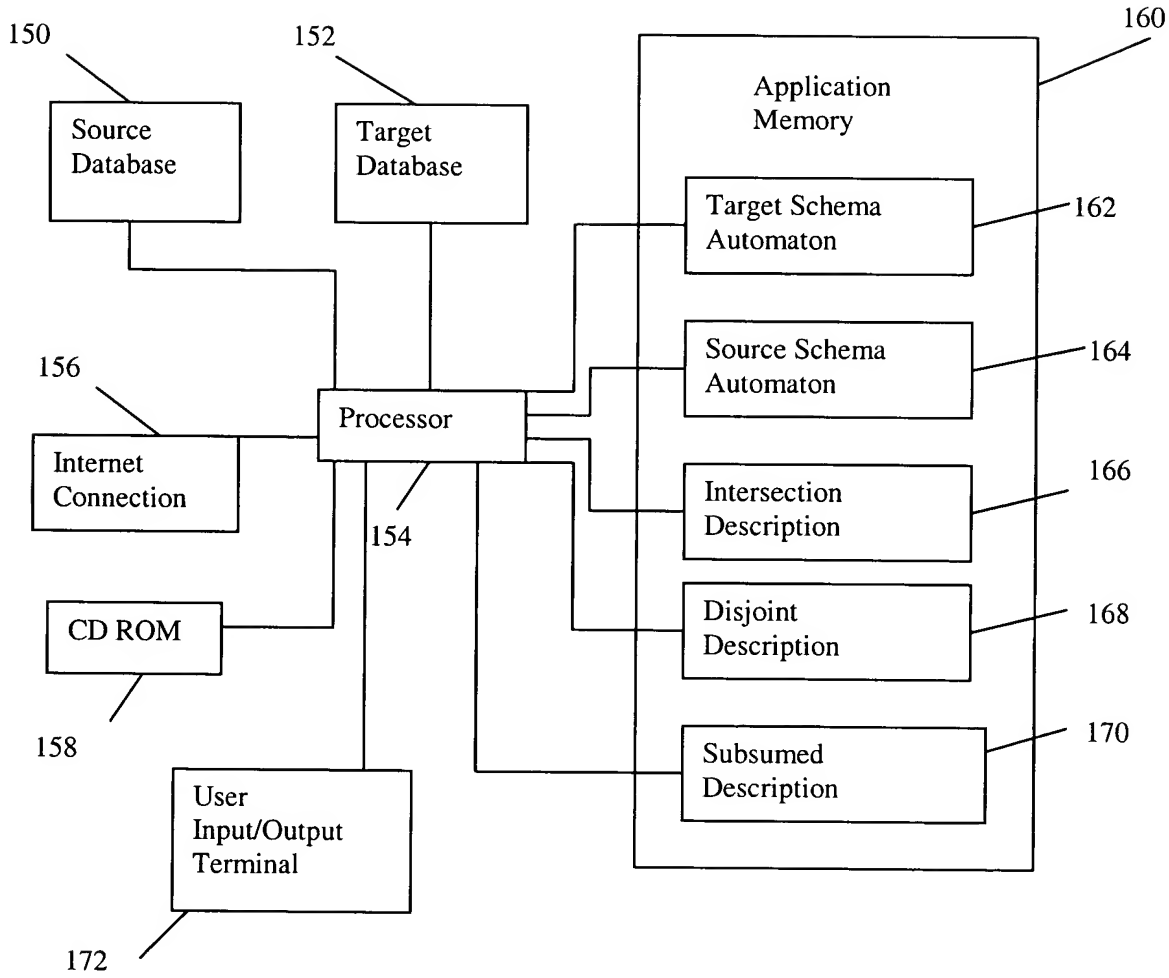
150
152
160
Source Database
Target Database
Application Memory
Target Schema Automaton
162
Source Schema Automaton
164
156
Processor
Internet Connection
154
166
Intersection Description
Disjoint Description
168
CD ROM
158
Subsumed Description
170
User Input/Output Terminal
172

Fig. 8